REMARKS

The claims previously of record have been cancelled and replaced with the new claims 27-48 set forth above. The new independent claims 27, 37 and 43 are derived from the previous independent claims 1, 12 and 21 respectively, and accordingly applicant will discuss the issues raised by the rejections of claims 1, 12 and 21 in the context of the new independent claims.

Claims 12-20 stand rejected under 35 USC 112, first paragraph. The new claim 37 takes account of this rejection and it is believed that all claims comply with the requirements of 35 USC 112, first paragraph.

Claims 13-16, 18 and 19 stand rejected under 35 USC 112, second paragraph. The new claims ** take account of this rejection and it is believed that all claims comply with the requirements of 35 USC 112, second paragraph.

The independent claims 1, 12 and 21 stand rejected under 35 USC 103 over Altberg et al. (US6353928) when considered in combination with Glover (US6052780).

Altberg describes a computer system that includes a shared library DLL and an installer module. Whenever an application is executed by the computer system, the application calls the shared library DLL in order to determine whether files required by the application are present on the hard disk of the computer system. The location of the required files is predetermined and is known by the shared library DLL and the installer module. If a file required by the application is missing, the installer module is executed (col. 5, lines 49-55 and 66-67 of Altberg). The installer module determines whether there is sufficient space on the hard disk to install the required file. If sufficient space is available, the installer module installs the required file to a predetermined location on the hard disk. Once installed, the application is then executed (col. 6, lines 3-5 and 27-29 of Altberg).

Glover describes an executable program that stores digital information in encrypted form. When the executable program is executed by a computer, the digital information is decrypted and stored in a secure area. In a first embodiment, the executable program decrypts the digital information directly into a protected memory area. The protected memory area is protected by the operating system of the computer such that it cannot be copied or accessed by other unauthorised processes (col. 3, lines 52-61 and col. 8, lines 56-66 of Glover). In a second embodiment, the executable program first extracts the encrypted digital information (referred to by Glover as hidden information) and stores the encrypted digital information as one or more phantom files in a phantom directory. The executable program then loads a device driver, which is responsible for decrypting the phantom files on demand. When the operating system requests access to the digital information, the device driver decrypts the relevant phantom file and delivers the decrypted digital information to the operating system (col. 4, lines 9-27 and col. 10, lines 1-47).

Claim 27 clearly indicates that the program, the at least one encrypted sub-routine, and the decryption routine all form part of the same executable application.

Neither Altberg nor Glover describes a self-contained executable application that comprises a program, at least one encrypted sub-routine required by the program, and a decryption routine operable to decrypt the sub-routine when access is required by the program.

Altberg teaches a computer readable medium having an executable application recorded thereon, the executable application comprising a program. However, the executable application (labelled 205 in Figure 2 of Altberg) does not comprise an encrypted sub-routine or a decryption routine. Instead, sub-routines required by the application are stored unencrypted on the hard disk of the computer at predetermined locations. If a particular sub-routine is not present, the installer module installs the relevant sub-routine to the predetermined location.

Glover teaches an executable application that stores a computer program in an encrypted form. When the executable application is executed, the computer program is decrypted and stored in a secure area ready for access by the operating system of the computer.

A person of ordinary skill in the art seeking to combine the teachings of Altberg and Glover would be taught that the executable application of Altberg (referred to hereafter as the primary application) may be provided in an encrypted form as taught by Glover. In particular, the person of ordinary skill in the art would provide the primary application as an executable application, the executable application storing the primary application in encrypted form and decrypting the primary application when the executable application is executed.

The person of ordinary skill in the art might also consider storing the sub-routines of Altberg in an encrypted form as taught by Glover. In particular, the person of ordinary skill might consider storing each sub-routine on the computer at a predetermined location as an executable application, the executable application storing the sub-routine in encrypted form and decrypting the sub-routine when access to the sub-routine is required.

Contrary to the contention made by the examiner in item 18 of the office action, the person of ordinary skill in the art would not, however, be taught that an encrypted sub-routine might somehow be incorporated within an unencrypted program. As noted above, the person of ordinary skill in the art would provide the primary application of Altberg as an encrypted executable and might additionally provide each sub-routine as a separate encrypted executable. The person of ordinary skill in the art would not, however, be taught that a sub-routine might somehow be provided in encrypted form alongside an unencrypted program.

Altberg specifically teaches that the primary application is provided separately from the sub-routines. Indeed, it is an essential feature of the system described by Altberg that the sub-routines are stored separately from the primary application at predetermined locations, such that the sub-routines can be used by different applications. It would therefore be contrary to the very teachings of Altberg to include a sub-routine within the primary application.

Further, a person of ordinary skill in the art contemplating incorporating a sub-routine within the primary application would not arrive at the claimed invention of claim 27.

A person of ordinary skill in the art minded to append a sub-routine to a primary application would naturally arrive at a single self-contained application. Following the teachings of Glover, the person of ordinary skill would then provide the self-contained application as an encrypted executable. Importantly, the entire contents of the self-contained application would be encrypted. The person of ordinary skill in the art would not consider encrypting only the sub-routine of the self-contained application whilst leaving the primary application unencrypted. Indeed, there is no teaching in either Altberg or Glover as to how an executable application comprising an unencrypted primary application and an encrypted subroutine might be achieved. Accordingly, even if it were regarded as obvious to append a sub-routine to a primary application, the person of ordinary skill in the art would still fail to arrive at the invention of claim 27 since both the primary application and the sub-routine would be encrypted to form a single encrypted executable.

In summary, the person of ordinary skill following the teachings of Altberg and Glover would provide the primary application as an encrypted executable and the sub-routines as separate encrypted executables stored at predetermined locations. The person of ordinary skill would not, however, provide a single executable application that comprises both the primary application and the required sub-routines. Even if it were deemed obvious to provide a single executable that includes both the primary application and the required sub-routines, both the primary application and the sub-routines would be encrypted. There is simply no teaching in either Altberg or Glover that would enable the person of ordinary skill to provide a single executable application that comprises a program, an encrypted sub-routine and a decryption routine that decrypts the sub-routine when access is required by the program.

It is therefore submitted that claim 27 is not rendered obvious in view of Altberg et al, alone or when taken together with Glover.

Independent claims 12 and 21 have been revised in a manner consistent with claim 27 to arrive at the new independent claims 37 and 43. It is therefore submitted that, for the reasons provided above, claims 37 and 43 are not rendered obvious in view of Altberg et al and Glover whether taken singly or in combination.

Additionally, claim 37 is directed to a system having a first store means storing the executable application and a second store means storing sub-routines. In this embodiment, the executable application includes loading means that identifies which sub-routines are required by the program. If any of the required sub-routines are stored on the second store means then the loading means loads these sub-routines for use by the program. If, however, a required sub-routine is not stored on the second store means then the loading means decrypts an encrypted copy of the sub-routine contained within the executable application. This decrypted copy of the sub-routine is then used by the program. Accordingly, with the system of claim 37, sub-routines already stored by the system are loaded and used by the program. If, however, a particular sub-routine is missing, the required sub-routine is decrypted and loaded instead.

Importantly, the executable application comprises all the sub-routines required by the program. Consequently, the executable application provides a fully-functioning program that makes use of shared sub-routines, where available, that are already present on the system.

As already noted above, neither Altberg nor Glover teach or suggest a self-contained executable application that includes an unencrypted program, all sub-routines required by the program stored in encrypted form, and a decryption routine that decrypts the sub-routines should they be required by the program. Additionally, the decryption routine described by Glover does not selectively decrypt the encrypted digital information according to whether or not the digital information already exists on the computer. Instead, when the encrypted executable is executed, the decryption routine decrypts the digital information without any regard to the contents of the computer. Accordingly, if, as suggested by the examiner in item 18 of the office action, the person of ordinary skill were to encrypt the primary application of Altberg using the method taught by Glover, the person of ordinary skill would fail to arrive at an executable application that includes a decryption routine that (i) determines whether sub-routines required by the primary application already exist on the computer and (ii) decrypts encrypted sub-routines only in the event that one or more of the required sub-routines are missing.

It is therefore submitted that claim 37 is not rendered obvious by Altberg et al and Glover, whether taken singly or in combination.

Whilst the remaining claims are dependent upon either claim 27, claim 37 or claim 43, and are therefore patentable by virtue of their dependencies, consideration will now be give to some of the dependent claims.

In item 19 of the office action, the examiner states that it is inherent from Altberg that the predetermined locations of the sub-routines are placed in an address table in order that the shared library DLL and installer module know in which location to look for a particular subroutine. It is not disputed by the applicant that the system described by Altberg must include a look-up table of the locations of each sub-routine. However, claim 29 of the present application indicates that the decryption routine decrypts the sub-routine and subsequently makes an entry in the address table. Conversely, the installer module of Altberg does not make an entry in a table. Instead, the installer module inspects the look-up table in order to determine the predetermined location of the sub-routine. If the sub-routine is not present, the sub-routine is then installed at the predetermined location specified by the look-up table. The installer module therefore relies upon the look-up table to store the location of each sub-routine. At no time does the installer module actually make an entry into the table.

The examiner also makes reference to col. 9, lines 47-53 of Glover in item 19 of the office action. The passage of Glover refers to a particular embodiment in which the encrypted executable stores more than one block of digital information. Each block of digital information may include begin and end tags such that the decryption routine can successfully decrypt each block of digital information. Alternatively, rather than using tags, the encrypted executable may include a look-up table that stores the locations of each of the encrypted blocks of digital information within the encrypted executable. When the executable is executed, the decryption routine accesses the look-up table to determine the location (i.e. the beginning and end) of each block of encrypted digital information. Without the look-up table, the decryption routine would not know where one block of information ends and the next block of information begins, and consequently decryption would fail. It is therefore essential that the contents of the table remain in tact and unchanged. It will, of course, be immediately apparent that the table is not modified by the decryption routine. In particular, the decryption routine does not decrypt a block of information and then store the location of the block of information within the table.

It is therefore submitted that claim 29 is not rendered obvious in view of Altberg and Glover, whether taken singly or in combination. The same is true of claim 39, which recites the same features as claim 29.

In item 37 of the office action, the examiner alleges that claims 7-8, 18-19 and 25 are rendered obvious in view of Altberg and Glover when taken together with Shen (US6611850).

Shen describes a backup method in which a file stored on a first storage device is copied to a second storage device, so as to provide a backup copy of the file. The backup copy of the file may be encrypted or compressed. Shen clearly teaches that the backup copy of the file should be stored separately from the original copy of the file. Moreover, the backup copy should be stored on a separate storage device from that of the original copy of the file.

The person of ordinary skill in the art would therefore be taught by Shen that any file to be backed-up should be stored separately from the original file. In contrast, claims 33, 41 and 47 indicate that the executable application comprises an unencrypted copy of the program and an encrypted copy of the program. The provision of an unencrypted copy and an encrypted copy of a program within a single executable application is simply not taught nor suggested by Shen. Indeed, the person of ordinary skill in the art would be clearly instructed by Shen that any backup copy of the primary application of Altberg should be stored as a separate file on a different computer readable medium.

It is therefore submitted that the invention claimed in claims 33, 34, 41, 42 and 49 is not disclosed or suggested by Altberg, Glover and Shen, whether taken singly or in combination.

In view of the foregoing, applicant submits that all claims currently of record are allowable.

Respectfully submitted,

John Smith-Hill
Reg. No. 27,730

SMITH-HILL AND BEDELL, P.C.
16100 N.W. Cornell Road, #220
Beaverton, Oregon 97006
Tel: (503) 574-3100
Fax: (503) 574-3197
Docket: FORR 2275
Postcard: 11/05-27

Certificate of Mailing

I hereby certify that this correspondence is being deposited as first class mail with the United States Postal Service with sufficient postage in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, on this 17th day of November, 2005.

John Smith-Hill